

SYSTEM AND METHOD FOR ASSESSING THE  
SECURITY POSTURE OF A NETWORK  
USING GOAL ORIENTED FUZZY LOGIC DECISION RULES

*Ins  
at  
B1*  
B1 at >

Field of the Invention

This invention relates to the field of networks, and more particularly, this invention relates to the field of assessing security vulnerabilities of  
5 networks.

Background of the Invention

Information systems and computer network infrastructures currently under development are now  
10 being built with consideration for what constitutes an acceptable risk (or adequate protection). System assets, such as the hardware, software and system nodes of a computer network, must be protected to a degree consistent with their value. Additionally, these  
15 assets must be protected only until the assets lose their value. Any security features and system architecture should also provide sufficient protection over the life of the processed data. To assess whether or not any risk associated with a network is  
20 acceptable, a security engineer typically gathers all pertinent information, and then analyzes the risk associated with the network.

Risk analysis is a complex and time consuming process, which is necessary to determine the exposures  
25 within a network and their potential harm. As an example, when analyzing the security risks in a computer network, the security engineering typically follows the following steps:

2) Identify vulnerabilities of assets. This step typically requires imagination in order to predict what damage might occur to the assets and from what sources. The three basic goals of computer security are ensuring secrecy, integrity and availability. A vulnerability is any situation that could cause loss of one of those three qualities.

4) Compute any uncovered cost per year (expected annual loss) by determining the expected cost of each incident.

6) Project annual savings of control.

Today, the rapid evolution of technology and proliferation of computers with increased power mandate the use of commercial-off-the-shelf (COTS) hardware and software components for cost effective solutions. This strong dependence on COTS implies that commercial grade security mechanisms are sufficient for most applications. Security architectures, therefore, must be structured to build operational, mission-critical computer systems with relatively weak COTS components. Higher assurance components can be placed at community

There are some design tools, i.e., software

1) Tools that work from documented vulnerability databases and possibly repair known vulnerabilities. Tools of this type are vendor-dependent for database updates, either through new product versions or by a subscription service. Examples from this category include ISS' Internet Scanner, Network Associates, Inc.'s CyberCop and Harris' STAT.

3) Tools that examine a particular aspect of the system, such as the operating system or database management system, but ignore the other system components. SATAN, for example, analyzes operating system vulnerabilities, but ignores infrastructure components such as routers.

The use of multiple tools from a variety of vendors for a single computer network analysis is a labor-intensive task. Typically, a security engineer will have to enter a description or representation of the system (network) multiple times in multiple formats. The security engineer then must manually

analyze, consolidate and merge the resulting outputs from these multiple tools into a single report of a network's security posture. Afterwards, the security engineer can complete the risk analysis (calculating  
5 expected annual loss, surveying controls, etc.), and then repeat the process to analyze alternatives among security risks, system performance, mission functionality and the development budget.

Also, none of these tools use an aggregate  
10 "snapshot" approach to the system with a "drill down" or layered approach to facilitate how one addresses risk at various layers (network, platform, database, etc.) of the system. These tools provide little assistance to system designers when analyzing  
15 alternatives among security risk, system performance and mission functionality. Instead, a "risk solution" is provided that addresses the particular aspect of risk that a given tool was designed to calculate. To develop a comprehensive risk assessment, a security  
20 engineer would have to become proficient in the use of several tools and manually correlate the resulting outputs.

One aspect of successful risk analysis is a complete and accurate accumulation of data to generate  
25 system models used by the analysis tools. Many current risk analysis tools depend on surveys filled out by users, system operations personnel, and analysts to acquire the data for development of a system model used in the analysis. Alternatively, a tool can actively  
30 scan a computer network to test various vulnerabilities against system components.

However, these methods have drawbacks. Textual or survey-based knowledge solicitation techniques are labor intensive and potentially tedious  
35 for the analyst. Many of the existing tools reuse the same information to analyze different aspects of the system security. It would be more advantageous to use

000000-000000

5  
10  
15

## Summary of the Invention

20

25  
30  
35

5 vulnerability posture of the network.

10 network vulnerability programs and importing via an

20 In still another aspect of the present invention, a computer program resides on a medium and can be read by a program and comprises instructions to cause a computer to create a system object model database representing a network and which supports the  
25 information data requirements of disparate network vulnerability analysis programs. A computer program causes the computer to import only the required data from the system object model database to each respective network vulnerability analysis program and  
30 analyze the network with each network vulnerability analysis program to produce data results from each program. The results are stored with the common system model database within a data fact base. The computer program also creates instructions to cause a computer  
35 to apply goal oriented fuzzy logic decision rules to the data fact base to determine the vulnerability posture of the network.

15           Data fact base stores the results obtained  
from respective network vulnerability analysis programs  
after analyzing the network and the common system model  
database and a fuzzy logic processor applies goal  
oriented fuzzy logic decision rules to the fact  
20 database by the use of a plurality of fuzzy expert  
rules for merging results from the network  
vulnerability analysis programs and determining the  
vulnerability posture of the network.

Other objects, features and advantages of the present invention will become apparent from the detailed description of the invention which follows, when considered in light of the accompanying drawings

30 in which:

FIG. 2 is another schematic block diagram of  
35 a network showing an identified vulnerability located  
by the system and method of the present invention.

5           FIG. 4 is another schematic block diagram of  
the architecture of the present invention showing the  
fuzzy logic analysis.

FIG. 6 is another high level schematic block diagram of the data processing system of the present invention.

FIGS. 8A and 8B show open windows that provide data resolution in the establishment of the system object model database.

FIG. 10 is a graphical user interface showing various reporting options for the security posture of the network.

FIG. 12 is a schematic block diagram of the data fusion used in the data processing system and method of the present invention.

35           FIG. 14 is another block diagram showing  
basic processing steps and components used in the fuzzy



logic processing of the data processing system and method of the present invention.

FIG. 15 is a block diagram showing basic components used in the fault tree analysis (DPLf) for evidence accumulation and fuzzy evidential reasoning rules.

FIG. 16 is a block diagram showing an object/class hierarchy.

FIG. 17 is a block diagram showing the system class diagram of the present invention.

#### Detailed Description of the Preferred Embodiments

FIG. 1 illustrates an example of a conventional network 100 having internal servers 102 that connect to an external router 104, communication network 105, and firewall 106. An internal router 108 is connected to the firewall 106, branch office 107, and connected to internal LAN network components 110 and a remote-access server 112 and remote user 114.

Using the example of FIG. 1, frequent problems found on networks include hosts, such as the internal servers 102, which run unnecessary services, for example, a denial of service and anonymous FTP or misconfigured web servers that could be an internal server, for example, CGI scripts, anonymous FTP and SMTP. The internal LAN's 110 could include unpatched, outdated, vulnerable or default configured software and firmware and weak passwords. LAN's could also include improperly exported file sharing services, such as NetWare file services and NetBIOS. The internal LAN 110 could also include misconfigured or unpatched windows NT servers and problems caused by a lack of comprehensive policies, procedures, standards and guidelines. A remote-access server 112 could have unsecured remote-access points and the external router

104 could have information leakage through services, such as SNMP, SMIP, finger, roosers, SYSTAT, NETSTAT, TELNET banners, Windows NT TCP 139 SMB (server message block), and zone transfers to non-named server hosts.

5 It could also have inadequate logging, monitoring and detecting capabilities. The branch office 107 could have a misappropriated trust relationship such as RLOGIN, RSH, or REXEC. The firewall 106 could be misconfigured or have a misconfigured router access  
10 control list.

Although these network problems are only an example of common problems found on networks 100, there are many other problems that could occur, as is well known to those skilled in the art.

15 The present invention is advantageous because the system and method of the present invention allows the vulnerabilities of a network system to be identified. The software of the data processing system and method can be located on a user terminal 120, as  
20 shown in FIG. 2, showing an identified vulnerability of a node 112 connected in the internal LAN 110. For purposes of description, the data processing system and method of the present invention can be referred to as a Network Vulnerability Tool (NVT), i.e., a tool a user  
25 uses to determine network vulnerabilities and risks.

The data processing system forming the NVT of the present invention can be loaded on a Pentium PC platform running Windows NT. This type of platform can provide a low cost solution and support a large variety  
30 of assessment tools, also commonly referred to as network vulnerability assessment or risk analysis programs throughout this description. These network vulnerability analysis programs typically are the standard COTS/GOTS programs known by security  
35 engineers, and include HP Open View, which allows network automatic discovery or manual network modeling;

008020 5920568

5

19

35

5

10

15

When a network system description has been completed, the NVT of the present invention represents

and stores the description in an object/class hierarchy, as shown as an example in FIGS. 16 and 17, as will be explained below. A single topological system object model supports the information data needs of the disparate network vulnerability analysis programs (tools). Fuzzy logic processing of the results allows correlation of the results from the programs into a cohesive vulnerability/risk assessment to obtain a vulnerability posture of the network, as shown in the graphical user interface at FIG. 10. The single representation of the system simplifies the use of multiple tools and eliminates redundant data entry. It also provides a foundation for addressing the problem of incomplete data for a given vulnerability assessment tool and future knowledge negotiation capabilities.

FIG. 3 illustrates at 130 an example of the overall network visualization tool (NVT), data processing system of the present invention, where three network vulnerability analysis programs (tools) are illustrated as ANSSR 132, ISS Internet scanner 134, and RAM 136. The system and method of the present invention creates a system object model database (Network Model DB) 138 that represents a network and supports the information data requirements of the network vulnerability analysis programs. The system object model database 138 represents a single representation of the assessed system or design, and addresses the need for a single internal representation of a network to provide data to the network vulnerability analysis programs.

This model 138 uses object oriented (OO) methodology to provide an extensible set of components in a class hierarchy that can be combined to represent a network. The class hierarchy provides a means of defining components with shared common traits, while

retaining the specifics that distinguished it from other components. In addition to an implicit hierarchical relationship, object oriented techniques provide a containment mechanism in which an object can contain a reference to any object, including itself. This provides a flexible mechanism for representing any physical or logical entity. Also, object oriented representation lends itself to ready modification and extension and is ideal for an information assurance arena where changes and new technologies arise daily.

As shown in FIG. 3, filters 140 are associated with each of the network vulnerability analysis programs 132, 134, 136 and allow only that data required by a respective network vulnerability programs to be exported to the tool (program). The filters are a C++ base class that provide a set of virtual methods to allow data movement between the NVT system and a program. The filter also provides a means for the NVT to control execution of the tool and complete data needed by a tool. NVT views each tool as a filter, calling the appropriate method within the filter to perform the desired task, including initializing, running, importing data and exporting data. Each tool can have a concrete filter subclass and provide the means to define each method specifically for the tool, while still providing the generic and well-defined programming interface (API) to NVT. This allows all tools to be treated the same within NVT, allowing the addition and removal of tools without changing any of the existing NVT codes.

Establishing communication between DPL-f and NVT using the filter technology is straightforward. A DPL-f filter is tasked with the specifics of building and populating fault trees. As an analysis tool, a default tree can represent a node in a network as developed and provide a probability value for events such as denial of service, loss of data and data

compromise. Actually, DPL-f can be used as a final result tool.

The network is then analyzed with each network vulnerability analysis program to produce data results from each program. The data results are correlated to determine a security posture of the network. Network validation can occur through the fuzzy logic processing of the invention, as will be explained below, and the system GUI can have input to a user display.

An overview of the network is created as a model 142 by an automatic network discovery or manual entry 144, such as through HP Open View, and an appropriate filter 146 allows the system GUI 148 to display the network model as shown in FIG. 7 via an appropriate data input 150 to a user display 152. It is also possible to have a risk GUI 154 to assess visually the risk vulnerability, a log 156 of the risk/vulnerability report, a risk assessment 158 as part of the GUI 148, all through the network validation 160, using a plug-in or fuzzy rule set as will be described in greater detail below. Any incomplete data resolution 161 can also be handled.

FIG. 4 illustrates a high level block diagram similar to FIG. 3, showing the system object model database 138 that can be established and work in conjunction with an integrated application programming interface 162 to allow importing of data into the various tools 164, as illustrated as a model tool, discovery tool and information analysis tools that result in the overall system results database 166. An application programming interface 168 and a graphical user interface 170 work in conjunction with model database 138. An evaluation/assessment manager 172

(manager) works in conjunction with an application programming interface (API) 174 and graphical user interface (GUI) 176 to correlate data results with fuzzy logic processing, indicated by dotted lines 178, including expert correlation 180 and fuzzy inferences and evidential reason 182 to produce vulnerability results 184 and a graphical user interface (GUI) 186 for the correlated results. Although FIG. 4 represents a high level model showing an example of different components, it is only one example of one type of high level components that could be used with the NVT system and method of the present invention.

FIGS. 5 and 6 illustrate other examples of high level models showing basic components and processing steps of the data sources 200 (FIG. 5), together with the system picture 202, a per tool analysis 204, a multi-tool analysis 206, the tool-to-expert analysis 208, and report media 210. The tool-to-expert analysis 208 could include the DPL-f 208a as part of the fuzzy logic processing in a data fact base, and use with CERT notes 208b and an expert system 208c for expert correlation. Reports can be generated, including output as icons on a graphical user interface, text, an EXCEL spreadsheet, Access and Configuration, as known to those skilled in the art. FIG. 6 also illustrates another high level model similar to FIG. 5, where the tools used to form a complete system object model and fuzzy logic process could include the individual tool processing and the multi-tool correlation.

FIGS. 7-10 illustrate in greater detail a graphical user interface 220 that can be contained on a computer screen and used for interacting with the NVT and determining the vulnerability posture of a network. As illustrated, the graphical user interface 220 is a



standard type of Windows™ interface. A system design window 222 permits the display of network icons 224 forming a network map that is representative of the relationship among different network elements and nodes contained within a network. Respective network icons 224 are linked together in an arrangement corresponding to how the network elements nodes are interconnected within the network. As shown in FIG. 7, the network elements can be linked together via connection lines 226, showing the interconnection that exists among actual network elements and nodes. The system design window 222 shows on the left side an internetwork view 230 with two nodes and a network view 232 on the right hand side of the window to illustrate a map of the network model. A manager window 234 is opened and displays properties of network elements.

A select data sensitivity pop up window (box) 240 is user selectable through the menu options for selected network elements (FIG. 8A), and has user selected items for selecting the sensitivity of network elements. The sensitivity for data on any node (node 1 in the example shown in FIG. 8A) can be selected for unclassified, sensitive, confidential, secret, restricted secret or top secret with appropriate Okay, Random and Default buttons.

A select node configuration edit pop up window (box) 250 is shown in FIG. 8B and can have user selectable vulnerability profiles for selecting a vulnerability profile of a network element or node. FIG. 9 also shows the network model diagram with the central hub and the interconnected nodes. It is possible that a user can edit the manager window 234 entries, which also allows the network discovery to occur through appropriate selection of buttons.

Naturally, network icons can be selected and moved as necessary for editing and design alternatives.

After the security posture has been established through the system, icons representative of high risk network elements can turn colors, such as red, the hub 252. Other selected icons could turn yellow, indicative of a less severe risk node, such as the HP4 node 254 shown in FIGS. 7 and 9. It is possible that shaded areas around the node or portions of the network could be colored red or yellow indicative of higher risk vulnerability. It is also possible that the connection line could turn red or yellow to indicate a poor connection between elements.

FIG. 10 illustrates a vulnerability posture window 270 for displaying user readable icons indicative of vulnerable network elements and icons. The overall system model is shown as part of an open system design window. However, a spreadsheet 272 is illustrated and a NVT risk assessment chart 274 having slider bars for risk assessment. A risk analysis window 276 showing the top five risk analysis elements is also illustrated.

FIG. 16 shows in greater detail a class hierarchy with the Class Names 280 as public attributes and private attributes, the Aggregation 282 and Association 284 of Source 286 and Target 288 with Generalizations 290. FIG. 17 illustrates an example of a system class diagram with various components identified in the blocks. Naturally, FIG. 17 is only a system class diagram as is known to those skilled in the art and is an example of what can be used for the system and method of the present invention.

Referring now in greater detail to FIGS. 11-15, the goal oriented fuzzy logic decision making is illustrated. As shown in FIG. 11, the system model

database 138 and results 300 from the respective network vulnerability analysis programs are combined together using an applications programming interface and expert correlation to form a data fact base 302

5 through data fuzzification. Goal oriented fuzzy logic decision rules operate through fuzzy inference network rules 304 and fuzzy evidential reasoning rules 306 to determine the security posture of a network based on predetermined goals 308.

10           The fuzzy logic processing of the present invention uses data fusion, evidential reasoning and inference network techniques. As known to those skilled in the art, evidential reasoning is a technique in which facts are gathered that support and refute a  
15 given hypothesis. The result is the proof or rejection of the hypothesis with a certain degree of confidence. The fuzzy logic processing of the present invention uses evidential reasoning to accumulate evident from the system and tool findings for each criteria, thereby  
20 merging the system assessment data into a single point of reference, the conformance of the system to a particular criteria. By suppling a set of criteria for fusion, the system constrains the fusion problem and reduces the search base. Evidential reasoning has  
25 previously been used to perform level-one multi-sensor data fusion, and is a common global reasoning technique in fuzzy expert systems, such as the type of system known to those skilled in the art as fuzzyCLIPS, developed by NASA. The result is a set of fuzzy  
30 evidential rules whose purpose is to accumulate evidence for a given set of requirements. This resolves potentially conflicting, ambiguous and redundant data from expert correlation and draws conclusions with available data, even if it is  
35 incomplete.

003020 15920050

The accuracy of the result is contingent upon the quantity and quality of the data available and it may be necessary to perform additional refinement on the available data prior to the application of fuzzy logic processing, while also maintaining the probabilistic nature of the data. This refinement uses inference networks and provides a method of reasoning about probability using heuristics, thereby removing the need for extensive a priori knowledge. The relation between the goals and potential security metrics encourages cross fertilization. As known to those skilled in the art, the fuzzyCLIPS uses fuzzy facts, which can assume any value between 0 and 1. The result can be viewed as a two dimensional plot of a continuous function bounded vertically by 0 and 1.

Data fusion is used with the system object database, data results data fact base. Intelligence data fusion is a multi-level, multi-disciplinary-based information process to yield the integration of information from multiple intelligence sources (and perhaps multiple intelligence disciplines) to produce specific and comprehensive, unified data about an entity (its situation, capabilities, and the threat it imposes). Data fusion provides information based on the available inputs. The intelligence data fusion process is typically partitioned into four levels, described in Table 1 below.

TABLE 1. THE LEVELS AND PURPOSES OF THE INTELLIGENCE DATA FUSION PROCESS

	Data Fusion Level	Description
5	1 Object Refinement	<ul style="list-style-type: none"><li>• Transforms data into consistent frame of reference</li><li>• Refines and extends, in time, estimates of object position, kinematics or attributes</li><li>• Assigns data to objects to allow application of estimation process</li><li>• Refines the estimation of object identity</li></ul>
	2 Situation Refinement	<ul style="list-style-type: none"><li>• Develops description of current relationships among objects and events in the context of the environment</li><li>• A symbolic, reasoning process by which distributions of fixed and tracked entities and events and activities are associated with environmental and performance data in the context of an operational problem</li></ul>
	3 Threat Refinement	<ul style="list-style-type: none"><li>• Projects the current "situation" into the future and draws inferences about threats, vulnerabilities and opportunities for operations</li></ul>
	4 Process Refinement	<ul style="list-style-type: none"><li>• Monitors process performance to provide information for real-time control and long-term improvement</li><li>• Identifies what information is needed to improve the multi-level fusion product</li><li>• Determines the source specific data requirements to collect required information</li><li>• Allocates and directs the sources to achieve mission goals</li></ul>

10

As noted before, NVT combines multiple types of data, from multiple sources, with other contextual information to form an integrated view of a networked system's security posture. NVT provides a user with a simple expression of the vulnerability posture of a given system or system design, and enables them to perform "what if" analysis for functionality, performance, and countermeasure trades, for the purpose of refining and improving the system or system design.

20

In computer security engineering, sensors are the various vulnerability assessment and risk analysis

tools, along with the GUI to gather information, as needed, from the user. The resulting outputs from these tools take the form of both qualitative and quantitative data, in a variety of formats from  
5 different vendors. For computer security engineering, the objects of interest are the nodes in a network (computing system), i.e. the assets, including hardware, software and data. The situation of interest is an assessment of the weaknesses in the security  
10 system of a computer network segment that might be exploited to cause harm or loss of secrecy, integrity or availability.

Assessing the risk faced by a computing system involves an assessment of the threats faced,  
15 their likelihood of occurrence (exploitation), and the expected cost of the loss (or harm). Finally, the network (computing system) can be refined based on the results of cost-benefits analysis. This requires information on protective measures (controls or  
20 countermeasures) appropriate for particular vulnerabilities and their costs. The cost-benefit analysis seeks to determine if it costs less to use a control or countermeasure, or accept the expected cost of the loss. This leads to the development of a  
25 security plan to improve security of a computer network system.

Table 2 contains an example of a first partitioning of this data fusion process for computer security engineering that could be used with the  
30 present invention, with four processing levels, corresponding to the four levels found in Table 1. As illustrated in FIG. 12, inputs to this process would consist of the object model database 138, results from individual tools 132, 134, 136, and other contextual  
35 information. The different data fusion levels 1-4 are indicated generally at 320, 322, 324 and 326.

5

Data Fusion Levels		Description
1	Node Data Refinement	<ul style="list-style-type: none"> <li>• Transforms data into consistent frame of reference</li> <li>• Refinement of data at the network node-level (the objects for computer security data fusion)</li> <li>• Data from multiple tools - correlated (assigned to appropriate nodes) and possibly combined at each node</li> <li>• Refines the estimation of object identity - network node (workstation) is a system-of-systems, consisting of an OS, critical applications, a database and data</li> <li>• Vulnerability analysis at this level does not yet constitute situation assessment</li> </ul>
2	Network Segment Refinement	<ul style="list-style-type: none"> <li>• Refinement of the situation at the network segment-level (system-of-systems level)</li> <li>• Develops description of current relationships among objects (nodes) in the context of the environment (a network segment)</li> <li>• A symbolic, reasoning process by which information about entities (nodes, network segments) and environment are associated with evidence about computer security goals, requirements</li> <li>• Combining tool results at the network segment-level</li> <li>• The situation of interest is the assessment of the network segment's vulnerabilities or exposures</li> </ul>
3	Risk Refinement	<ul style="list-style-type: none"> <li>• Refinement of the exposures and their potential for harm (risk) within a computing system</li> <li>• Projects the current "situation" (state of the computer network system) into the future and draws inferences about threats, vulnerabilities and opportunities for operations</li> <li>• Based on vulnerabilities, concerns, context, cost, threats</li> <li>• Refinement of a system design with the identification of controls that reduce one or more vulnerabilities</li> <li>• Based on countermeasures, components, cost</li> <li>• Identifies what information is needed to improve the multi-level fusion product</li> <li>• Facilitate long-term improvement of the system</li> </ul>

While the data fusion used in the present invention provides a conceptual framework for addressing the problem of merging results from multiple vulnerability assessment and risk analysis tools, expert systems, inference networks and evidential reasoning are used to implement the fusion concepts and merge tool results. The flexibility of fuzzy decision technology, in particular, fuzzy expert systems, offers the means to address these problems. A primary benefit of a fuzzy expert system is its ability to use and assimilate knowledge from multiple sources.

Fuzzy logic provides the technique for representing and inferring from knowledge that is imprecise, uncertain or unreliable. Similar to traditional expert systems, a fuzzy expert system can represent knowledge in the form of a system of IF/THEN rules in which the antecedents, consequent, or both are fuzzy rather than crisp. Fuzzy logic is used to determine how well fuzzy facts match the rules, and to what degree this match affects the rule's conclusion.

In accordance with the present invention, an inference network is a hierarchy of heuristic rules that can propagate probabilities without requiring extensive knowledge of *a priori* probabilities (e.g. Bayesian networks). The heuristic rules can be developed using expert knowledge on how the probabilities propagate, allowing conclusions to be drawn with limited knowledge of *a priori* probabilities. This results in low-level discrete probabilities being accurately reflected in higher-level conclusions. Probabilities of low-level events (such as probability of password compromise based on lifetime) need to be part of any conclusions drawn on higher-level events (vulnerability of password).

Initial studies of NVT uses accumulation of evidence to modify a fuzzy-fact and represent the change in state required by the current system. This



5

10

30

35

5 phrased as "Goal-based Fusion".

15 used to gather evidence to contradict or support the  
hypothesis.

30 measure of how well auditing is implemented.

35 NVT into a unified report. In particular, FuzzyFusion™  
is developed to implement Levels 1 and 2 fusion.

FuzzyFusion™ is accomplished through the use of a fuzzy

expert system (Goal-Oriented Fuzzy Logic Decision Rules) using FuzzyCLIPS, which combines the outputs of the various tools, user concerns about system risks and vulnerabilities, and expert understanding of the

5 results of each tool and how these fit into a larger information system security picture. Thus, NVT users obtain a simple expression of the security posture of a given computing system, or system design, and can perform "what if" analysis for functionality,

10 performance, and countermeasure trades.

FIG. 14 illustrates the NVT FuzzyFusion™ component architecture for implementing the first two levels of data fusion for computer security engineering. As the figure illustrates, the task of

15 modeling security expertise is partitioned into discrete tasks. Separation of Expert Correlation (Data Framework Merge Rules), Fuzzy Inference Network Rules, and Fuzzy Evidential Reasoning Rules addresses the problems of brittle expert systems and computational

20 explosion. It also segregates low-level data correlation and fusion from the resolution of ambiguous/conflicting data and the merging of results into one picture. This should result in fuzzy expert systems that are easier to maintain than one large

25 comprehensive system. Elements of this architecture are described below.

Data Fuzzification 310 converts the results from the individual vulnerability assessment and risk analysis tools 132, 134, 136 into fuzzy-facts, and

30 stores those along with the Common System Model (CSM), i.e., system object model database 138, into the (FuzzyCLIPS) Fact-Base 302. Individual tool results (after fuzzification) and the CSM 138 are exported for Expert Correlation processing 3310 (Data Framework

35 Merge Rules) to resolve system information and integrate tool output based on security expertise.

Expert opinion can be used to determine the specific fuzzy values attributed to the low-level events.

The Expert Correlation (Data Framework Merge Rules) 330 is a collection of fuzzy expert rules to perform node-level data refinement (Level-1) or network-segment refinement (Level-2). These rules correlate and consolidate the (fuzzified) outputs from the vulnerability assessment and risk analysis tools, using expertise from security engineers. These rules leverage extensive experience in security assessment to resolve low-level systems data and tool results. These rules resolve system information and integrate tool output. Expert Correlation Rule processing 330 can also transform low-level data from the CSM and tool results into high level conclusions. For example,

**IF auditing is on with these flags,  
AND the audit data is not backed up,  
THEN auditing is unreliable.**

20

Working from fuzzy-facts in the Fact-Base 302, a set of Level-1 fusion rules can consolidate the vulnerabilities for each node, resulting in a vulnerability rating for each node in the network. This rating can be imported back to NVT for display. Similarly, a set of Level-2 fusion rules can consolidate the vulnerabilities for each network segment, resulting in a vulnerability rating for each network segment. This can again be imported back for display.

The data is then subject to Fuzzy Inference Network Rules processing 304. It may be necessary to perform additional refinement on the available data prior to the application of Fuzzy Evidential Reasoning Rules 304, while maintaining the probabilistic nature of the data. This refinement will use inference networks, as known to those skilled in the art, which

provides a method of reasoning about probability using hueristics, thereby removing the need for extensive a priori knowledge.

Fuzzy Evidential Reasoning Rules 306 are a  
5 collection of fuzzy expert rules to merge individual  
tool results into a higher level assessment, from a  
systems-level perspective, of a network's security  
posture. These rules provide a mechanism for merging  
the CSM, tool results and the results from the Expert  
10 Correlation (Data Framework Merge Rules) 330 into a  
unified report. This also removes the necessity of  
dealing with incomplete and conflicting data from the  
forward-chaining expert system used in Expert  
Correlation.

15 Evidential reasoning use a technique in which  
facts are gathered to support and refute a given  
hypothesis. The result is the proof or rejection of  
the hypothesis with a certain degree of confidence.  
FuzzyFusion™ uses evidential reasoning to accumulate  
20 evidence from the Common System Model and tool findings  
for each criterion, thereby merging the computer  
network system assessment data into a single point of  
reference, the conformance of the system to particular  
criteria. By supplying a set of criteria for fusion,  
25 NVT constrains the fusion problem and reduces the  
search space, referred to earlier as goal-based fusion.  
The result will be a set of fuzzy evidential rules  
whose sole purpose is to accumulate evidence for a  
given set of requirements. This resolves the  
30 potentially conflicting, ambiguous and redundant data  
from Expert Correlation (Data Framework Merge Rules)  
330, and draws conclusions with the available data,  
even if it is incomplete. Obviously, the accuracy of  
the result is contingent upon the quantity and quality  
35 of the data available.

As noted before, the fuzzy logic processing is goal oriented. Goals for Evidence Accumulation processing 350 may be derived from a Security Requirements Database 352, a Computer Security Metrics Database 354, or a Vulnerability Database 356, such as a database composed of AFCERTs. Bounding fusion to pre-defined goals limits computation times. FuzzyFusion™ goals provide mechanism to obtain IA metrics.

10           The FuzzyFusion™ process has a number of advantages over traditional approaches. Crisp expert systems would require extremely large knowledge bases to encompass the necessary data and, yet, would still have a problem with incomplete data and conflicting  
15 results. Bayesian and probability networks require extensive and often unavailable a priori knowledge of probabilities. Algorithmic solutions do not fit the probabilistic and heuristic nature of the security problem.

20           Rete-based expert systems such as FuzzyCLIPS suffer from a geometric increase in execution time based on the number of rules and facts present in the system. This leads to breaking the analysis into subnetworks. FuzzyFusion™ will add subnetwork and  
25 scaling capabilities. The nodes for each subnetwork will be evaluated as a group, and then groups of subnetworks will be evaluated. Grouping the rules for each type of analysis into different modules will reduce the size of the Rete-network. In addition to  
30 decreasing execution time, this will also introduce a scalable method of analyzing networks that maps to the network model used by NVT.

As shown in FIG. 15, the other possible data spaces could include a threat knowledge database 360,  
35 cost database 362 as part of Level 3 fusion and a

This application is related to copending patent applications entitled, "**SYSTEM AND METHOD FOR**

Many modifications and other embodiments of the invention will come to the mind of one skilled in the art having the benefit of the teachings presented in the foregoing descriptions and the associated drawings. Therefore, it is to be understood that the invention is not to be limited to the specific embodiments disclosed, and that the modifications and embodiments are intended to be included within the scope of the dependent claims.